Differential Evolution with Improved Population Reduction

Ming Yang
School of Computer Science,
China University of Geosciences
430074, Wuhan, China
yangming0702@gmail.com

Zhihua Cai
School of Computer Science,
China University of Geosciences
430074, Wuhan, China
zhcai@cug.edu.cn

Jing Guan
Institute for Pattern Recognition and Artificial Intelligence,
Huazhong University of Science and Technology
430074, Wuhan, China
g_jing0414@yahoo.com.cn

Wenyin Gong
School of Computer Science,
China University of Geosciences
430074, Wuhan, China
cug11100304@yahoo.com.cn

ABSTRACT

In the Differential Evolution (DE), there are many adaptive DE algorithms proposed for parameter adaptation. However, they are mainly focus on the the mutation factor F and crossover probability CR. The adaptation of population size NP is not widely studied in the scope of DE. If reduce population size but not jeopardize performance of the algorithm significantly, it could reduce the number of evaluations for individuals and accelerate algorithm's convergence speed. This is beneficial to the optimization problems which need expensive evaluations. In this paper, we propose an improved population reduction method, considering the difference between individuals, and embed it into classic DE/rand/1/bin strategy, named dynNPMinD-DE. When population needs to reduce, select the best individual and the individuals with minimal-step difference vectors to form a new population. dynNPMinD-DE is applied to minimize a set of 13 scalable benchmark functions of dimensions D=30. The results show that compared with selecting better individuals and DE/rand/1/bin, dynNPMinD-DE can get better results on average, and the convergence becomes faster and faster as each population reduction.

Categories and Subject Descriptors

G.1.6 [**NUMERICAL ANALYSIS**]: Optimization —*global optimization, unconstrained optimization*

General Terms

Algorithms

Keywords

Differential evolution, adaptation, population size, reduction, difference vector

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

1. INTRODUCTION

Differential Evolution (DE) is a simple yet powerful evolutionary algorithm (EA) for global optimization introduced by Price and Storn [2]. Although adaptive DE algorithms have been proposed, they are mainly focus on the the mutation factors F and crossover probabilities CR. The adapting of population size parameter NP is not yet widely studied in the scope of DE. The main population adaptation methods are selecting the best new-size individuals and cloning the tbest individual [3, 4]. In [1], Brest et al. proposed a novel population size reduction method based on adaptive DE, named dynNP-DE. If reduce population size but not jeopardize performance of the algorithm significantly, it could reduce the number of evaluations for individuals and accelerate algorithm's convergence speed. This is beneficial to the optimization problems which need expensive evaluations. This article is to research the effect of population reduction on the classic DE/rand/1/bin and apply an improved method to it.

2. DE WITH IMPROVED POPULATION REDUCTION

In this section, we research the effect of population adaptation on the classic DE/rand/1/bin and apply an improved population size adaptation strategy based on the one in [1] to DE/rand/1/bin.

2.1 Poplulation Size Adaptation Strategy

$pmax$ denotes the number of different population size, and $pamx - 1$ reductioins need to be performed. NP_p ($p = 1, 2, \ldots, pmax$) is the different population size, and NP_1 is the initial population size. gen_p denotes the number of generations with poplulation size NP_p. dynNP-DE used an equal number of evaluatoins $\frac{maxnfeval}{pmax}$ for each population size. The number of generation gen_p with the population size NP_p is calculated as:

$$gen_p = \left\lfloor \frac{maxnfeval}{pmax \cdot NP_p} \right\rfloor + r_p, \quad (1)$$

where $r_p \geq 0$ is a small non-negative integer value, which is

Table 1: Experimental results of 30-dimensional problems f_1–f_{13}, averaged over 50 independent runs.

F	*Func. Eval.*	*dynNPMinD-DE* Mean (Std Dev)	*dynNP-DE* Mean (Std Dev)	*DE/rand/1/bin* Mean (Std Dev)
f_1	100000	**1.727E-010 (4.399E-010)**	2.189E-010 (7.783E-010)	1.720E+000 (5.581E-001)†
f_2	150000	**9.373E-010 (2.833E-009)**	9.767E-010 (2.317E-009)	2.315E-001 (7.443E-002)†
f_3	300000	**1.872E-004 (9.962E-004)**	2.846E-004 (8.454E-004)	3.601E+001 (1.309E+001)†
f_4	100000	**1.904E+000 (1.338E+000)**	1.951E+000 (1.129E+000)	9.072E+000 (1.305E+000)†
f_5	30000	**7.738E+001 (6.219E+001)**	8.690E+001 (1.302E+002)	5.882E+005 (1.906E+005)†
f_6	50000	**0.000E+000 (0.000E+000)**	**0.000E+000 (0.000E+000)**	1.923E+002 (4.955E+001)†
f_7	10000000	**3.436E-005 (1.325E-005)**	3.640E-005 (1.355E-005)	3.414E-004 (9.827E-005)†
f_8	200000	**-1.189E+004 (3.603E+002)**	-1.158E+004 (5.446E+002)†	-5.460E+003 (2.779E+002)†
f_9	200000	**2.190E+001 (6.986E+000)**	2.222E+001 (7.432E+000)	1.921E+002 (9.147E+000)†
f_{10}	200000	**9.784E-013 (1.900E-012)**	2.358E-012 (3.976E-012)	2.994E-003 (5.991E-004)†
f_{11}	70000	1.973E-003 (4.252E-003)	**1.580E-003 (3.547E-003)**	1.324E+000 (7.948E-002)†
f_{12}	150000	5.132E-017 (4.859E-017)	**3.828E-017 (1.844E-017)**	2.553E-003 (1.656E-003)†
f_{13}	150000	1.791E-016 (3.989E-016)	**8.994E-017 (1.365E-016)**	1.652E-002 (1.089E-002)†

† dynNPMinD-DE performs significantly better than the algorithm at a 0.05 level of significance by paired samples Wilcoxon signed rank test.

greater than zero when $maxnfeval$ is not divisable by $pmax$. The reduction is performed in the following generations:

$$G_R \in \left\{ gen_1, gen_1 + gen_2, \ldots, \sum_{p=1}^{pmax-1} gen_p \right\}. \quad (2)$$

And reduces population size by half at each reduction.

2.2 An Improved Population Reduction Method

Algorithm 1 A new population reduction method

Compute $d_{ij,G}$ for individuals:

$$d_{ij,G} = \sum_{k=1}^{D} \left| x_{i,k,G} - x_{j,k,G} \right|, \quad (3)$$

where $\mathbf{x}_{i,G}, \mathbf{x}_{j,G} \in P_G$ and $i < j$;

$P_{G+1} = \{\mathbf{x}_{best,G}\}$, where $\mathbf{x}_{best,G}$ is the best individual;

while $\left|P_{G+1}\right| < NP_{G+1}$ **do**

  For $\forall \mathbf{x}_i, \mathbf{x}_j \in P_G$, but $\mathbf{x}_i$ and $\mathbf{x}_j \notin P_{G+1}$ simultaneously, find individuals $\mathbf{x}_m$ and $\mathbf{x}_n$ with the minimal $d_{ij,G}$;

  if $\mathbf{x}_m \notin P_{G+1}$ **then**

    $P_{G+1} = P_{G+1} \bigcup \{\mathbf{x}_m\}$;

  end if

  if $\left|P_{G+1}\right| < NP_{G+1}$ and $\mathbf{x}_n \notin P_{G+1}$ **then**

    $P_{G+1} = P_{G+1} \bigcup \{\mathbf{x}_n\}$;

  end if

end while

In this section, we propose a new selection method named dynNPMinD-DE. P_G denotes the population set at G-th generation. When $G = G_R$, decrease the population size and the new population set P_{G+1} is generated using Algorithm 1. This strategy can make population closer and closer to the global optimum as evolution progresses.

3. EXPERIMENTS

In this section, dynNPMinD-DE is applied to minimize a set of 13 scalable benchmark functions in [5] when D=30. Parameters are set to be F=0.5, CR=0.9 and NP=200 for DE/rand/1/bin. $pmax$=4 and NP_{init}=200 as recommended in [1]. The numbers of function evaluations are set to be when dynNPMinD-DE reaches convergence.

Table 1 summarizes the average mean and standard deviation results of 50 independent runs. For clarity, the best results are marked in boldface. The comparison shows that dynNPMinD-DE performs better than DE/rand/1/bin and dynNP-DE.

4. CONCLUSIONS

In the late of evolution, small step of difference vectors in mutation is beneficial to generating better trial vectors. When population needs to reduce, dynNPMinD-DE selects the best individual and the individuals with minimal-step difference vectors to form a new population. So it can get smaller distance between individuals of population. The experimental results show that dynNPMinD-DE can get better results on average, and the convergence becomes faster and faster as each population reduction.

5. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No.61075063) and the Special Fund for Basic Scientific Research of Central Colleges, China University of Geosciences (Wuhan) (No.CUGL100230).

6. REFERENCES

[1] J. Brest and M. S. Maučec. Population size reduction for the differential evolution algorithm. *Applied Intelligence*, 29(3):228–247, 2008.

[2] R. Storn and K. Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.

[3] J. Teo. Exploring dynamic self-adaptive populations in differential evolution. *Soft Comput.: Fusion Found., Methodologies Applicat.*, 10(8):673–686, 2006.

[4] V. Tirronen and F. Neri. Differential evolution with fitness diversity self-adaptation. In R. Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, volume 193 of *Studies in Computational Intelligence*, pages 199–234. Springer Berlin / Heidelberg, 2009.

[5] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.*, 3(2):82–102, 1999.